

Figure 1

Prior Art

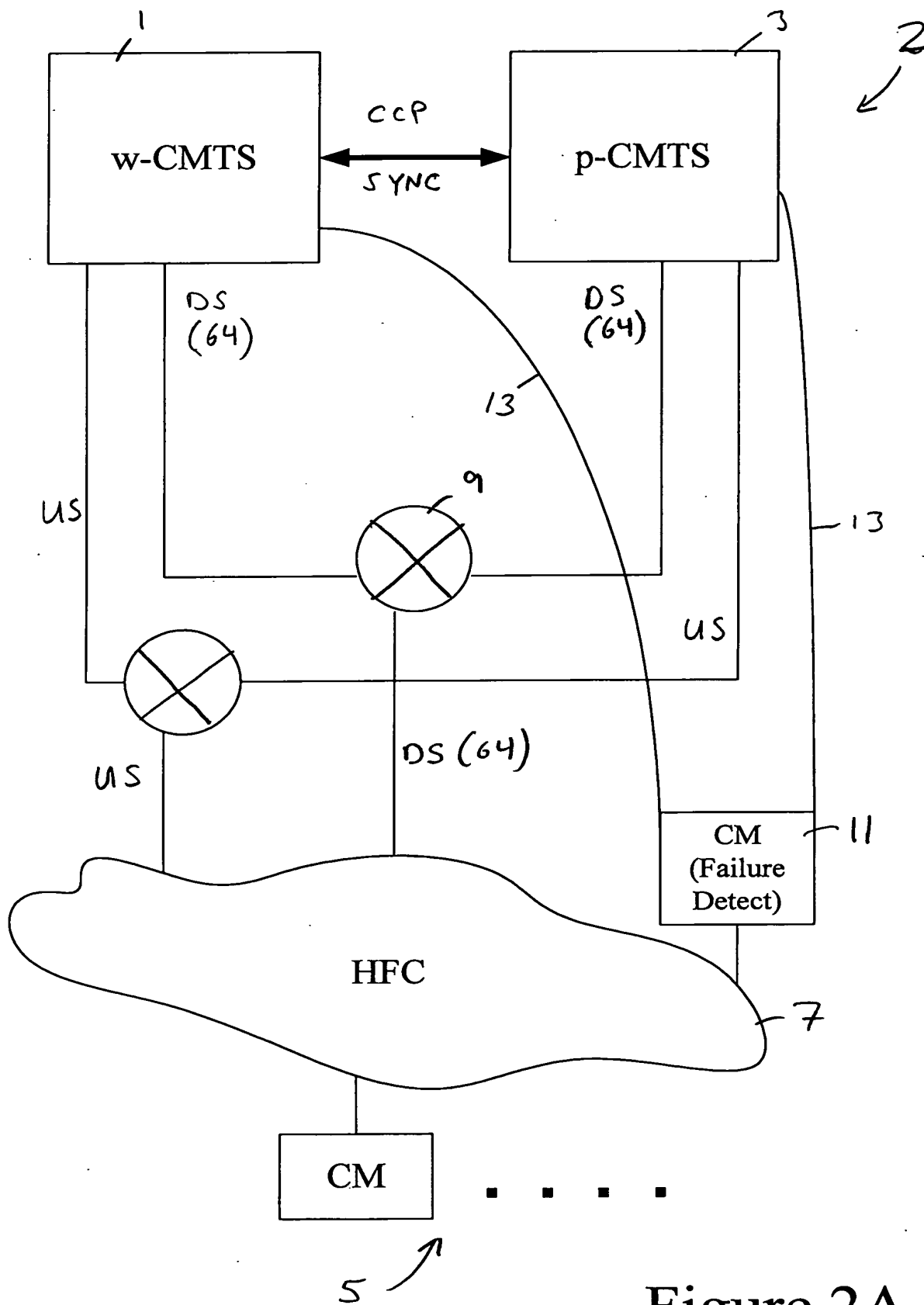


Figure 2A

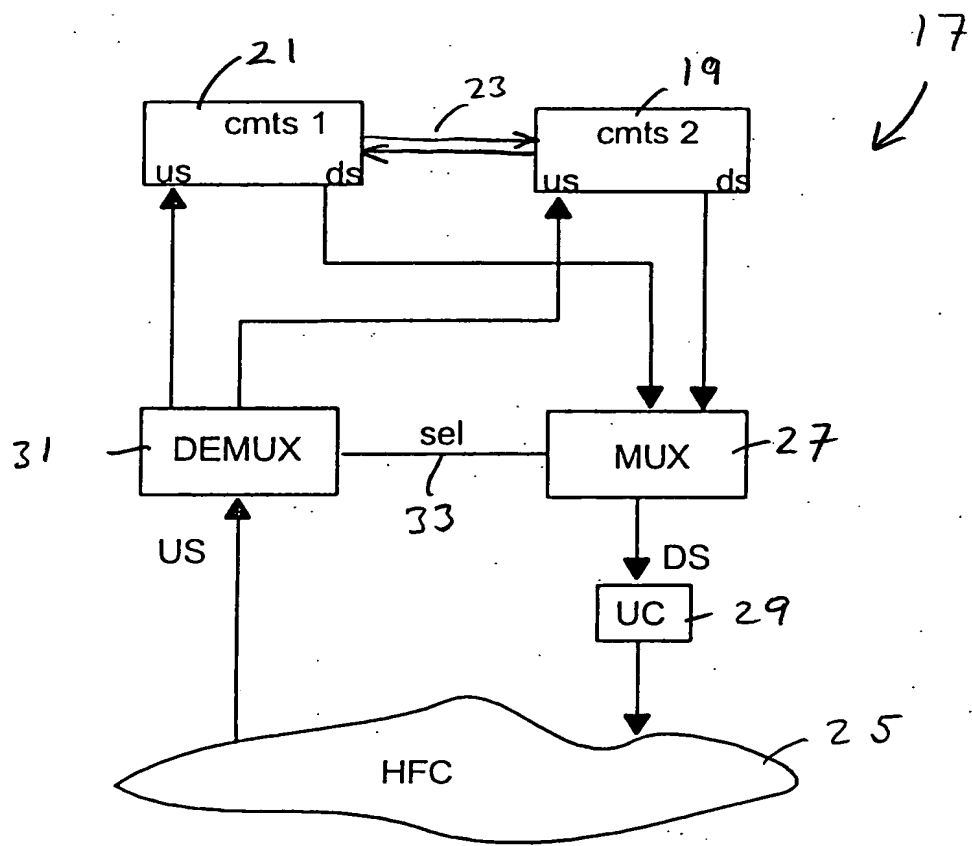


Figure 2B

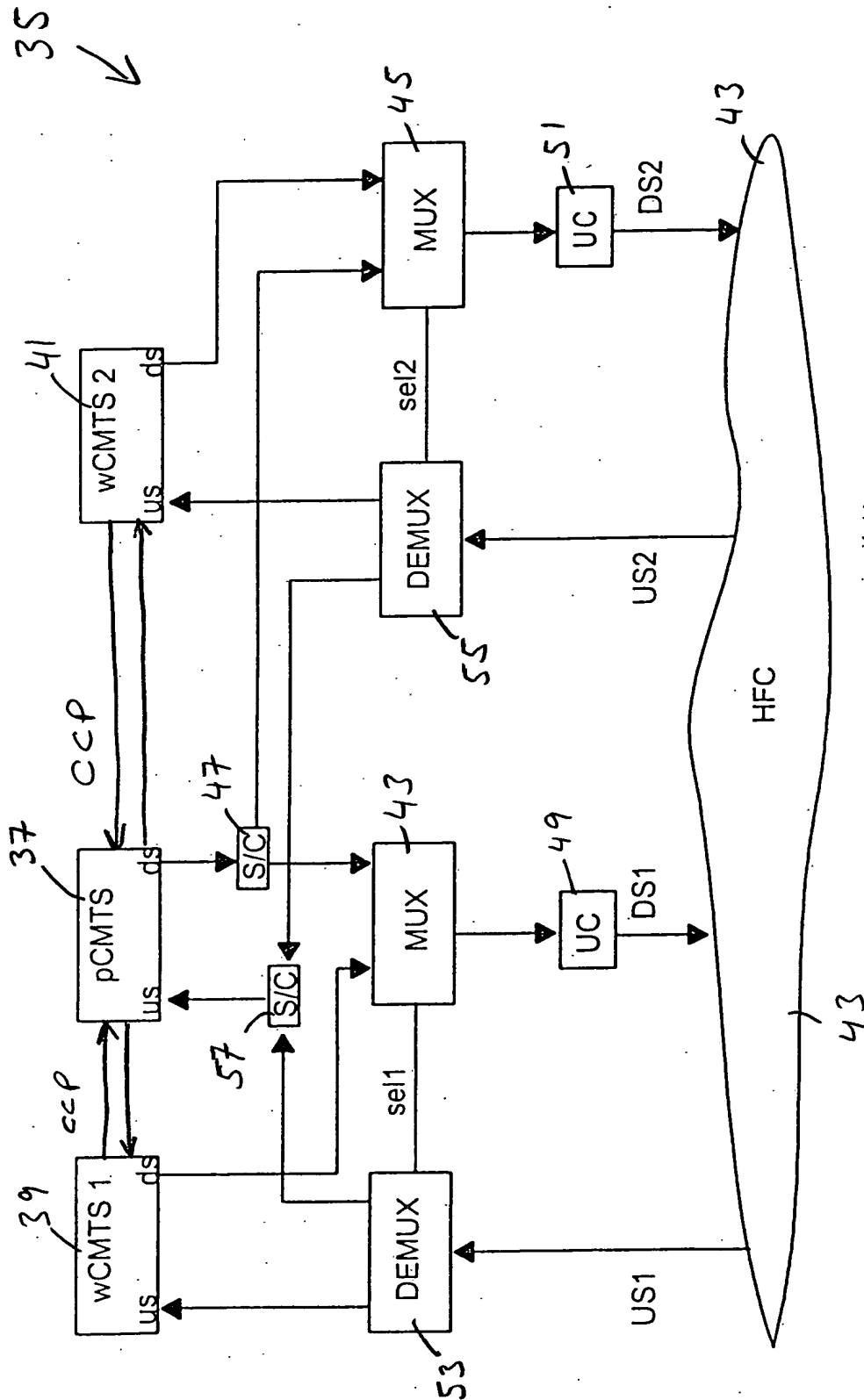


Figure 2C

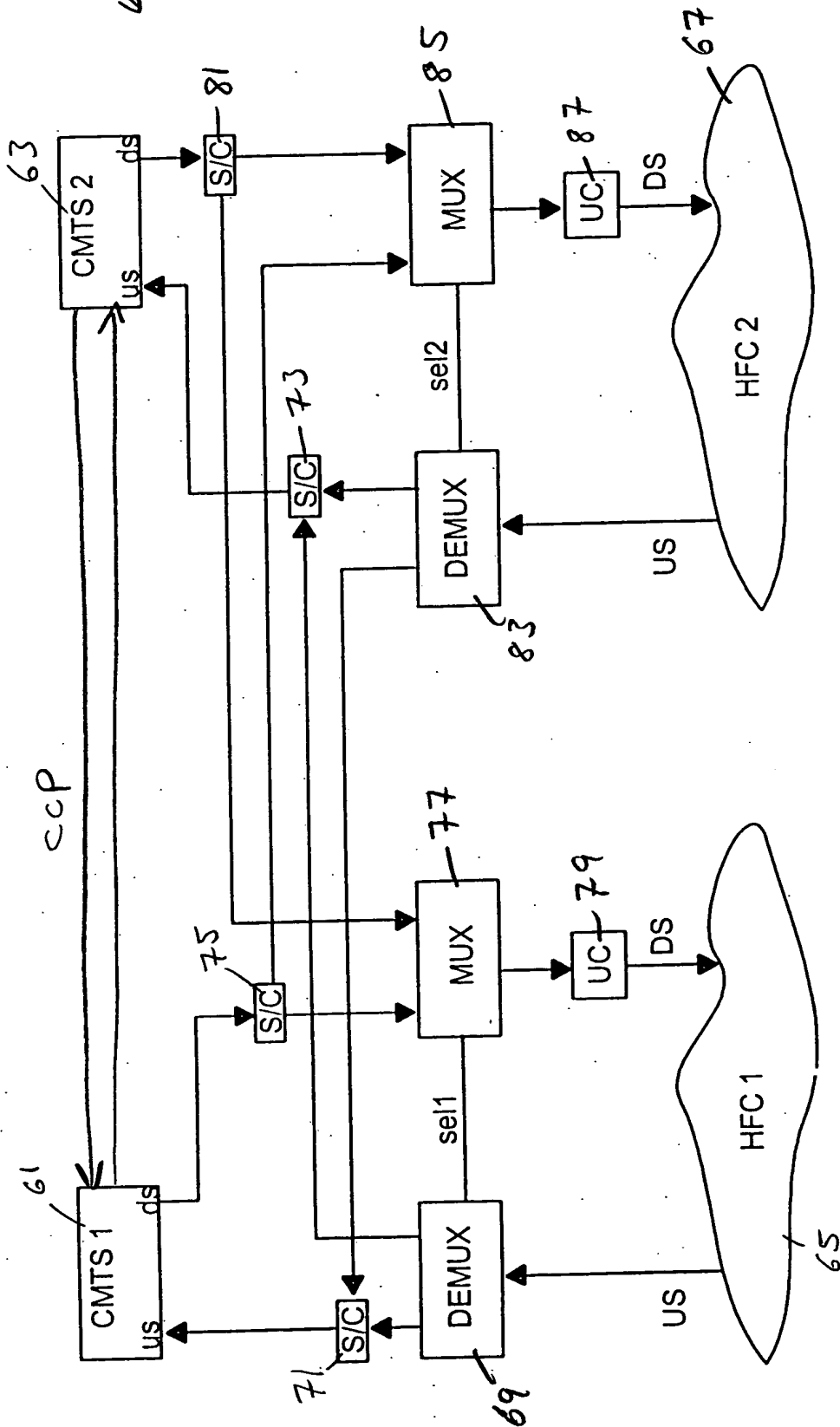


Figure 2D

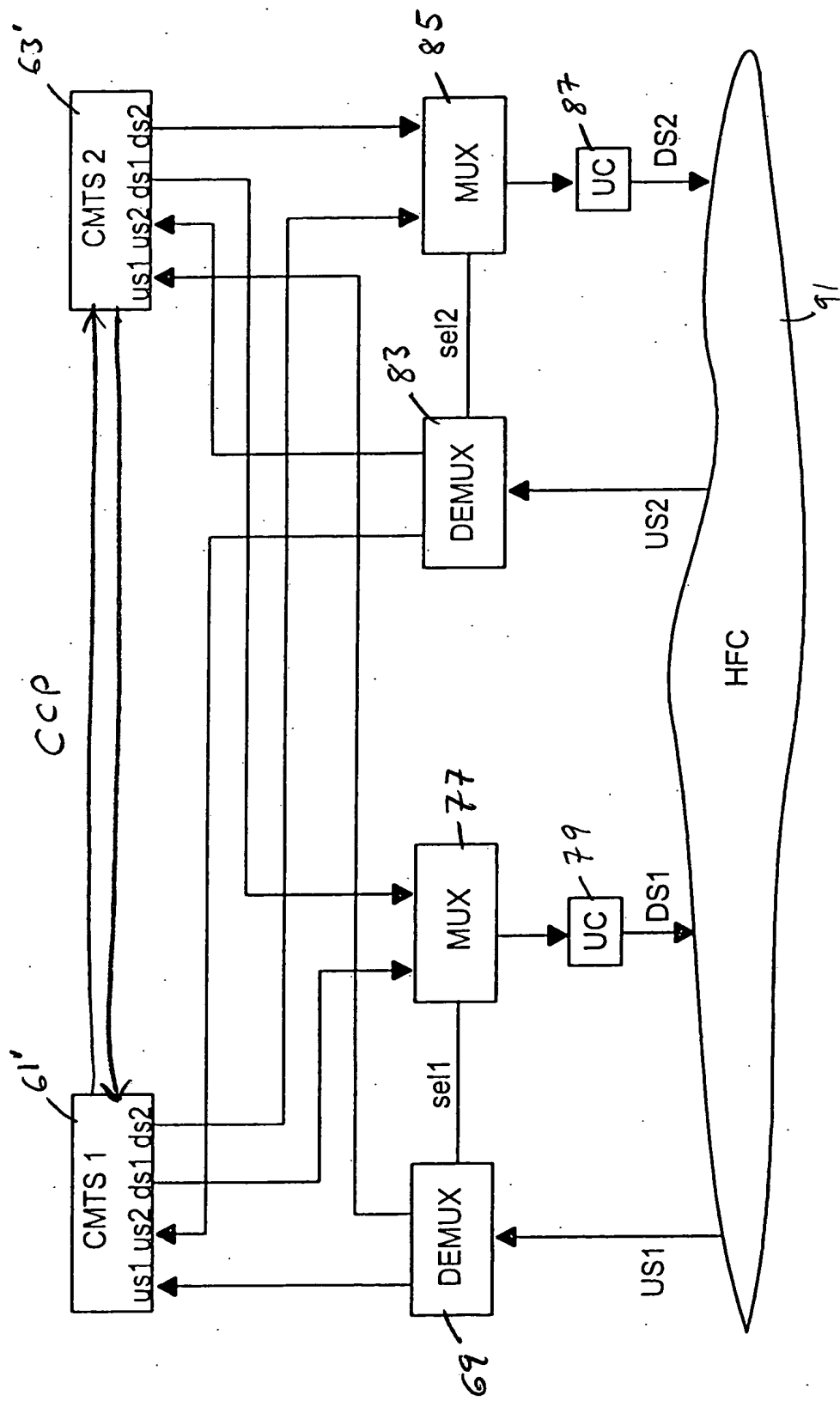


Figure 2E

301

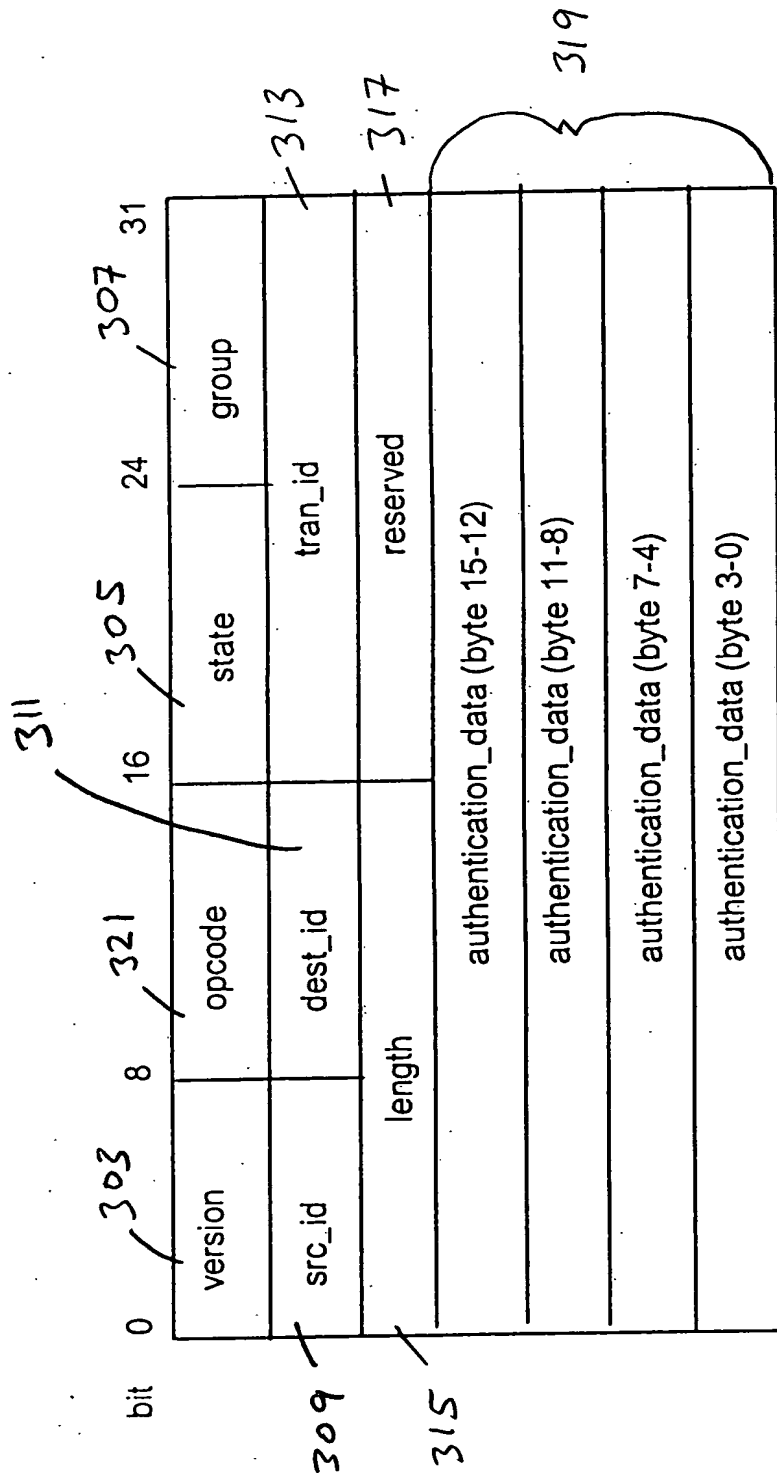


Figure 3A

323

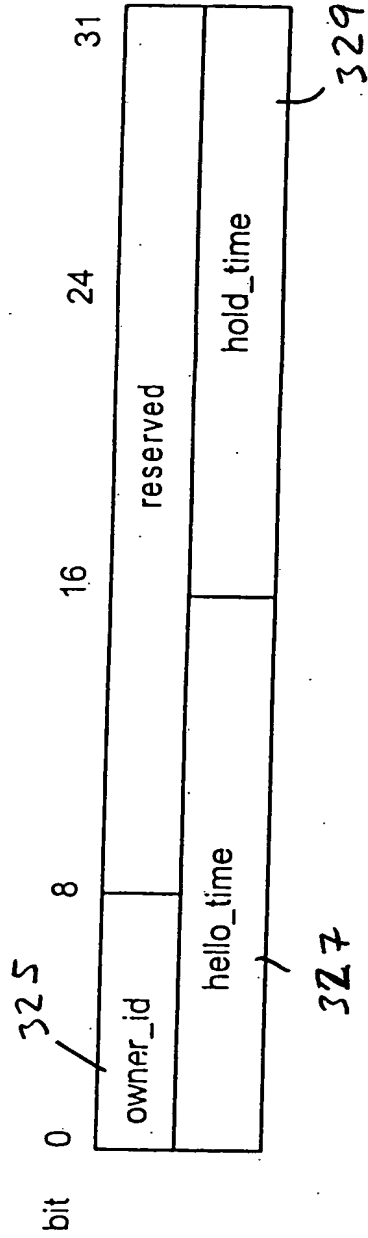


Figure 3B

331

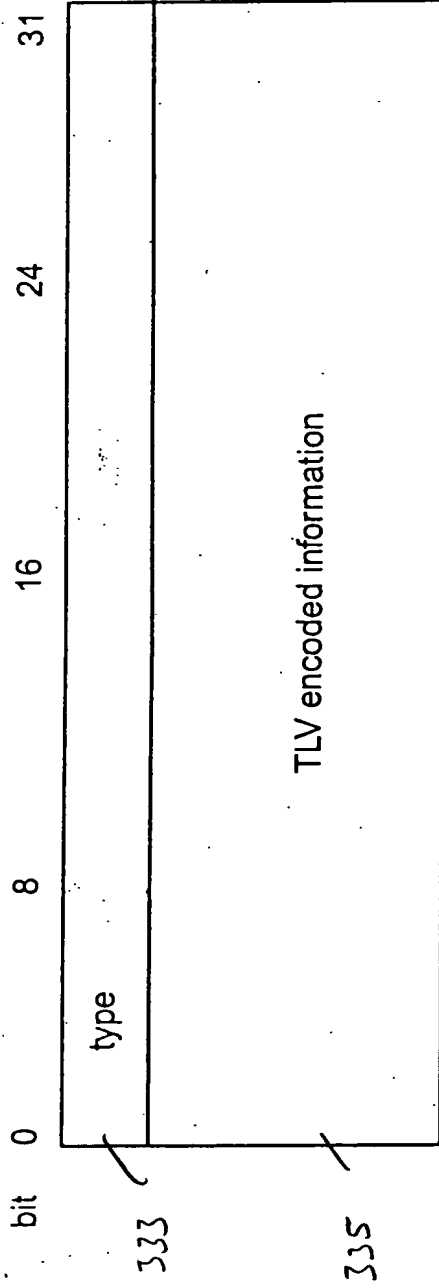


Figure 3C

Value Message	Name Message	Description
1	LOCKOUT	Teaching wCMTS tells pCMTS that it is not switchable.
2	UNLOCKOUT	Teaching wCMTS tells pCMTS that it is switchable.
3	RESYNC	Teaching wCMTS tells pCMTS that it is performing resync its entire database.
4	REG CM	Registration. Contains DOCSIS REG_REQ TLVs.
5	UCD	Upstream channel description. Contains DOCSIS UCD TLVs.
6	RNG	CM Ranging. Contains DOCSIS RNG_RSP TLVs.
7	DSA	CM Dynamic service add. Contains DOCSIS DSA_REQ TLVs.
8	DSD	CM Dynamic service delete. Contains DOCSIS DSD_REQ TLVs.
9	DSC	CM Dynamic service change. Contains DOCSIS DSC_REQ TLVs.
10	BPKM	CM Baseline privacy key management. Contains DOCSIS(8) TLVs.
11	SNA	Subnet add. Contains the MAC and IP address and IP mask for the sub-interface of CM and CPE to be added: 6 bytes MAC address, 4 bytes IP address, 4 bytes IP mask.
12	SND	Subnet delete. Contains the MAC and IP address and IP mask for the sub-interface of CM and CPE to be deleted: 6 bytes MAC address, 4 bytes IP address, 4 bytes IP mask.
13	SYNC	Time synchronization. Contains DOCSIS SYNC Timestamp.
14	~255	Reserved for future use.

Figure 4

State	wCMTS	Behavior
w_init	hccp->active = NULL; hccp->standbylist: 0 element;	transition state during initialization.
w_standalone	hccp->active != NULL; hccp->standbylist: 0 element;	forwarding traffic;
w_teach	hccp->active != NULL; hccp->standbylist: 0 element;	forwarding traffic; receive HELLO; send HELLO_ACK; send SYNC; receive SYNC_ACK;
w_learn	hccp->active = NULL; hccp->standbylist: 1 element;	receive HELLO; send HELLO_ACK; receive SYNC; send SYNC_ACK;

Figure 5A

State	pCMTS	Behavior
p_init	hccp->active = NULL; hccp->standbylist: 0 element;	transition state during initialization.
p_standalone	hccp->active != NULL; hccp->standbylist: n-1 elements;	forwarding traffic; send HELLO; receive HELLO_ACK; receive SYNC; send SYNC_ACK;
p_teach	hccp->active != NULL; hccp->standbylist: n-1 elements;	forwarding traffic; send HELLO; receive HELLO_ACK; receive SYNC; send SYNC_ACK; send SYNC; receive SYNC_ACK;
p_learn	hccp->active = NULL; hccp->standbylist: n elements;	send HELLO; receive HELLO_ACK; receive SYNC; send SYNC_ACK;

Figure 5B

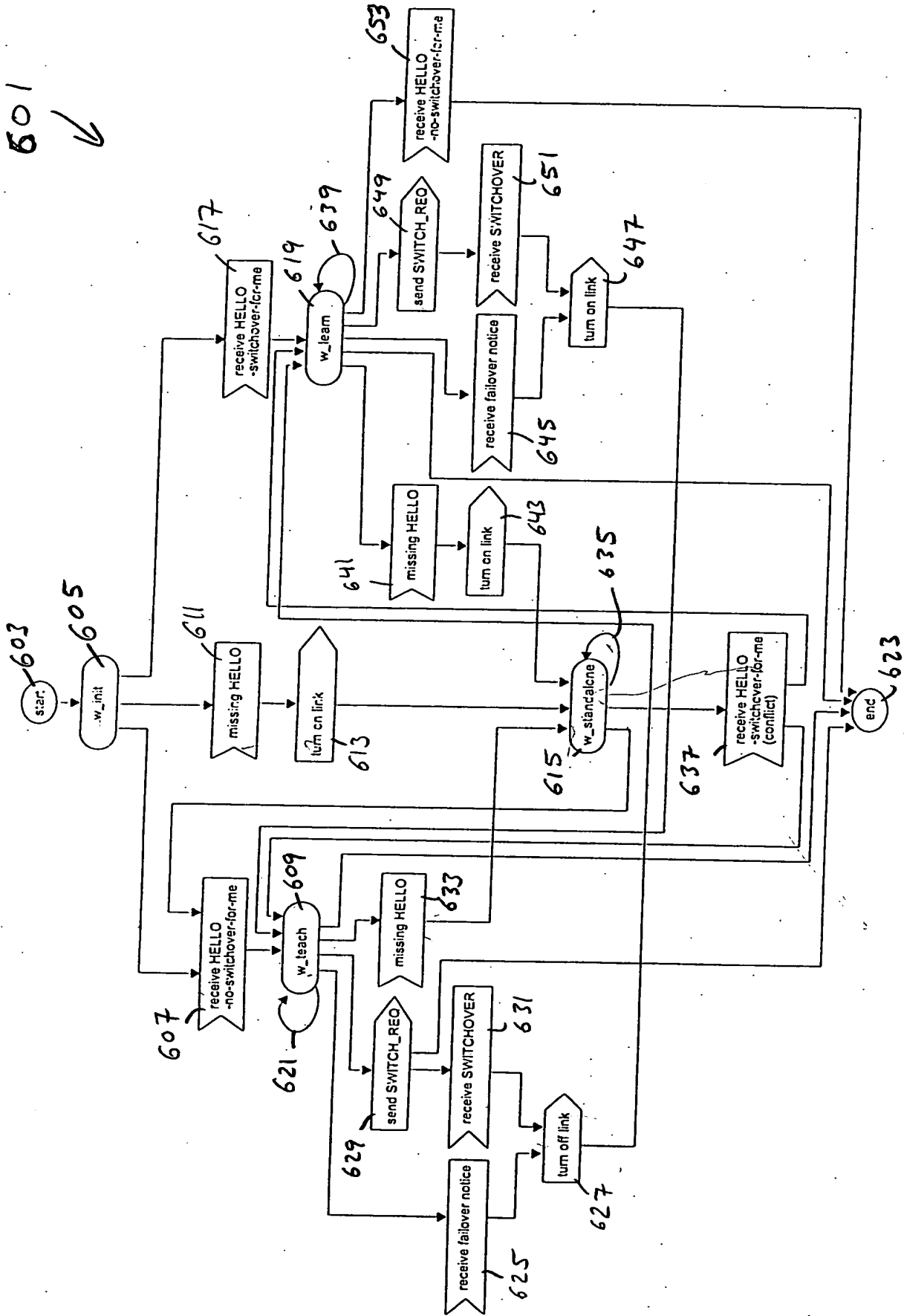


Figure 6

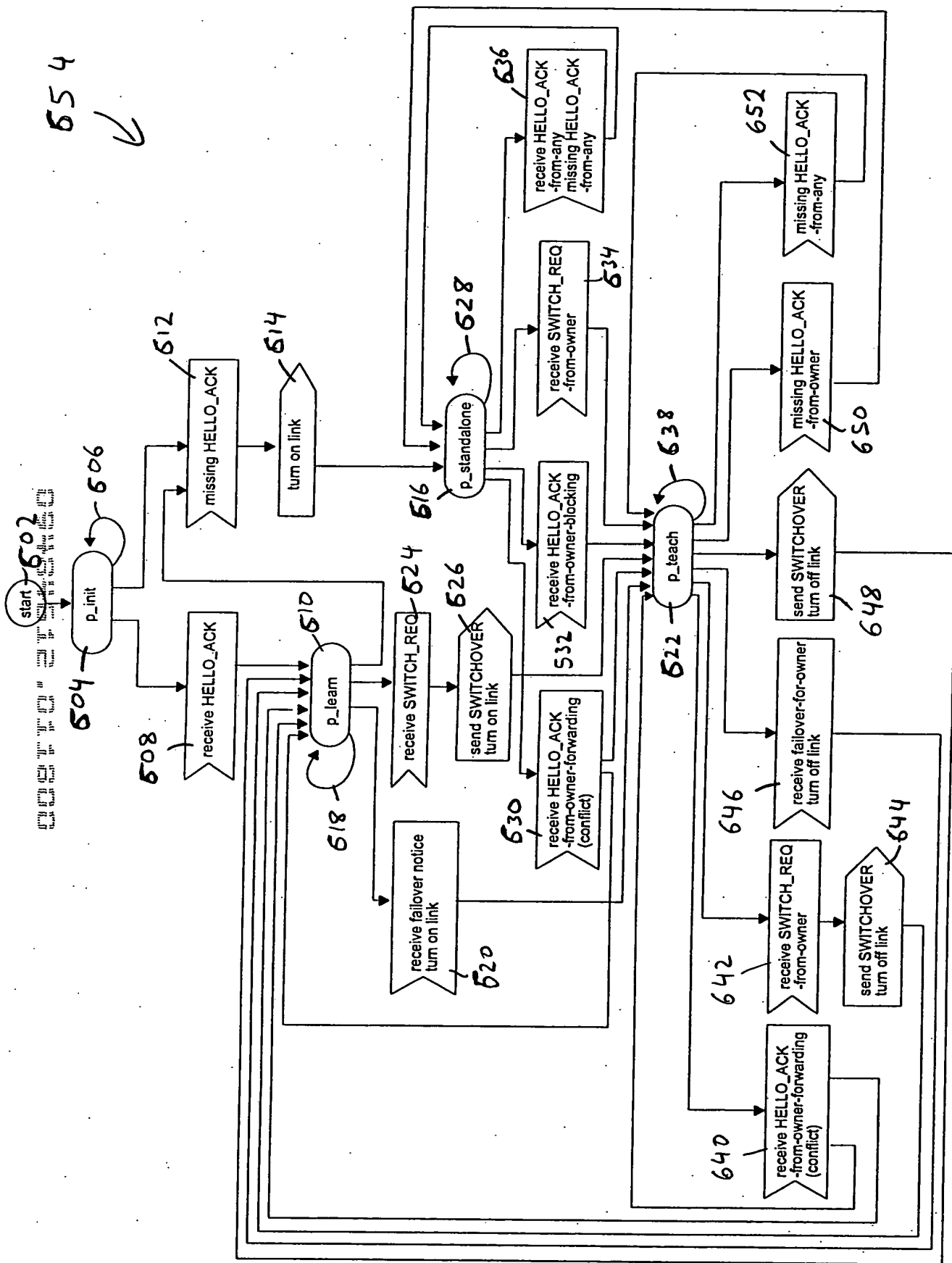


Figure 7

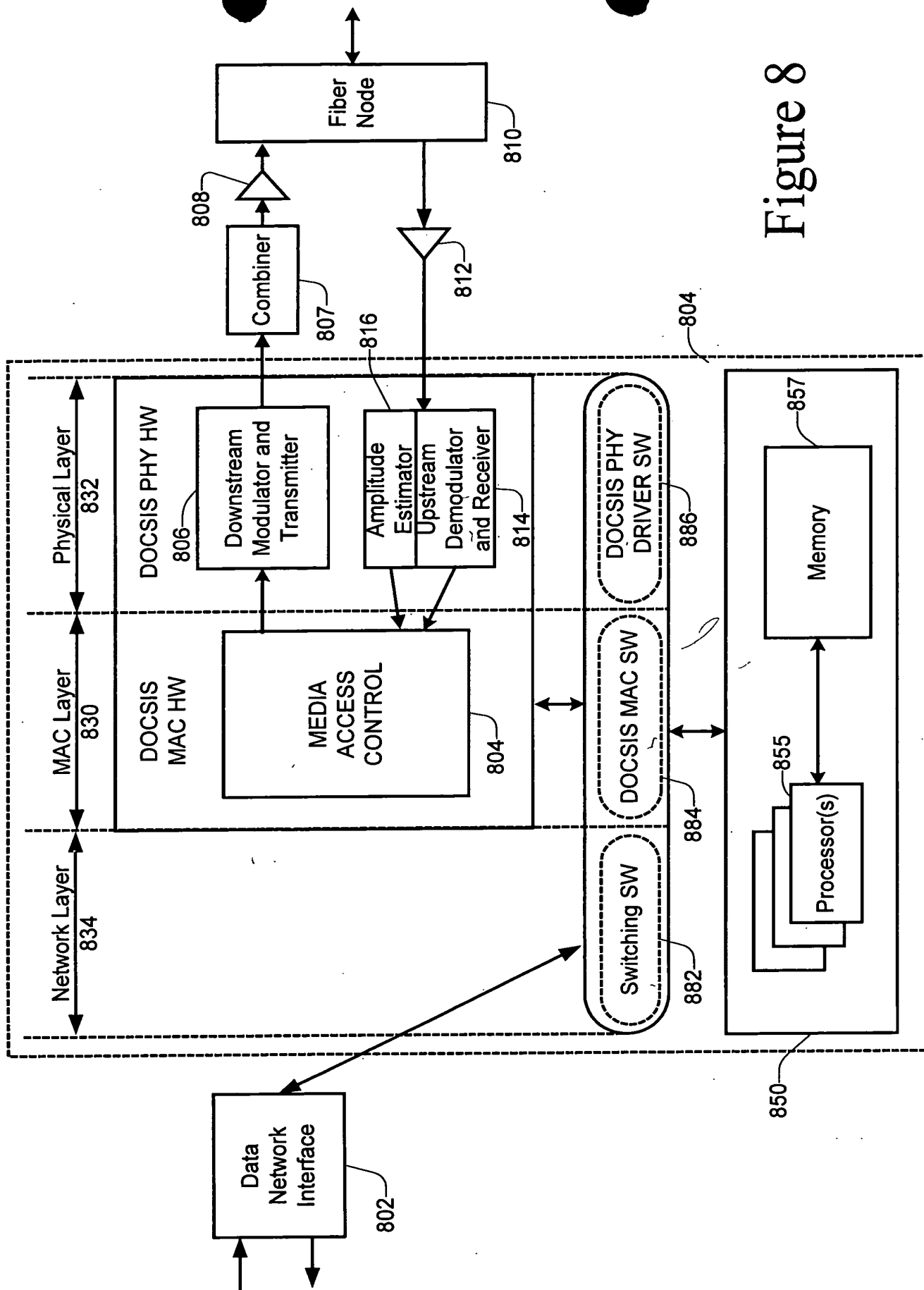


Figure 8

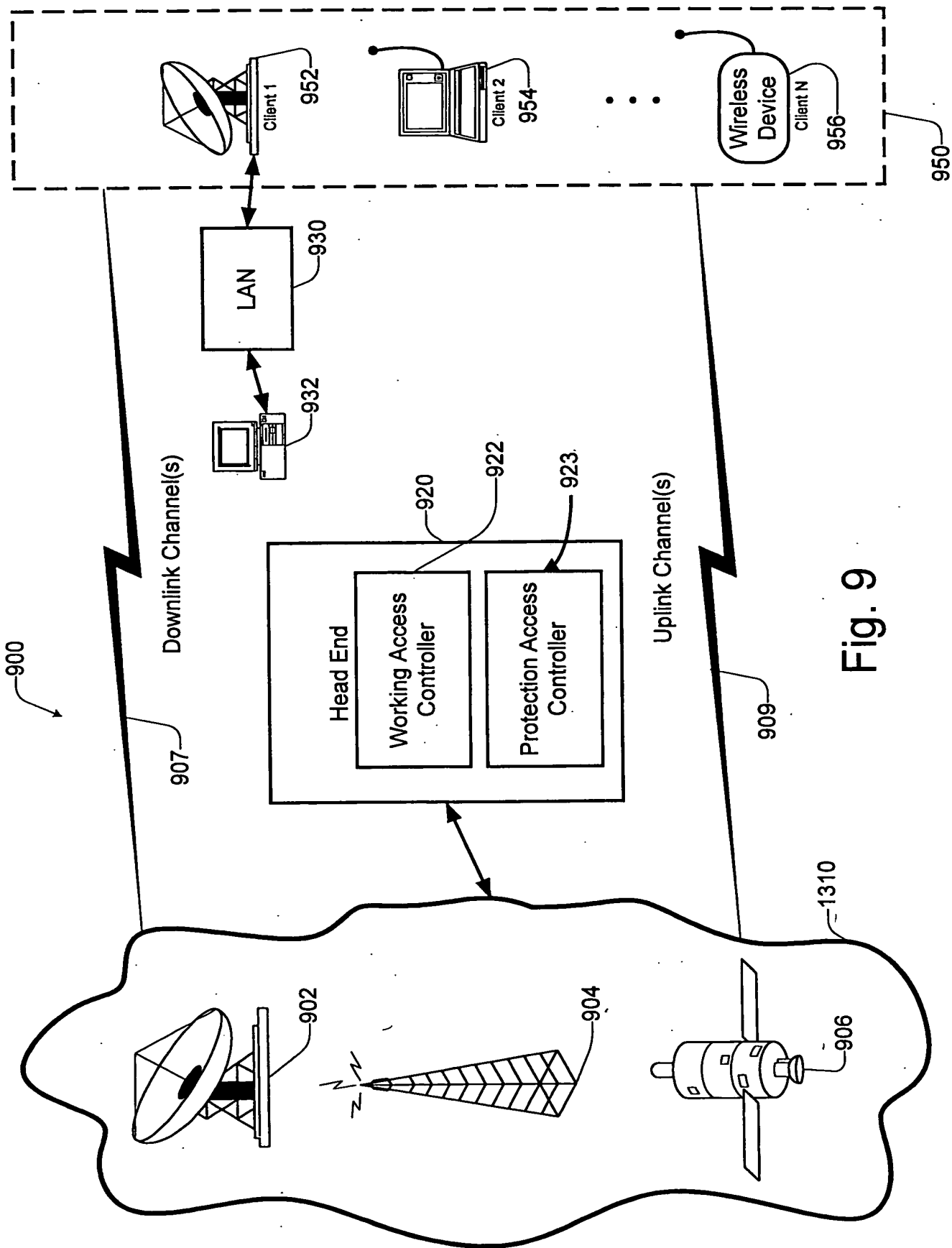


Fig. 9